ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

Topology Selection Method for CMOS Analog Circuits Using Machine Learning

[1] Stutee Pradhan, [2] Shashank M Rao, [3] Harshitha N, [4] Muralidhar Shanbhag

[1] [2] [3] [4] PES University, Bengaluru, Karnataka, India.
Corresponding Author Email: [1] stuteepradhan3004@gmail.com,[2] shashankrao1311@gmail.com,
[3] harshithan5559@gmail.com, [4] muralidhars@pes.edu

***Abstract*— Topology selection is an important part of analog circuit design and choosing the right topology manually is time-consuming. We present a new technique of topology selection for CMOS analog circuits. The technique involves predicting different amplifier topologies using a Machine Learning (ML) algorithm. The proposed implementation helps reduce errors and time spent on tedious, repetitive circuit designing tasks. It is aimed at reducing resource requirements and enhancing speed. For demonstration, we have focused on predicting 3 different amplifier topologies namely common drain, cascode amplifier and two stage OPAMP. 9 different performance parameters are used to distinguish different topologies. The topology prediction is performed using 5 Machine Learning algorithms and the performances of the algorithms are compared to find the best algorithm for topology selection.**

***Index Terms*—Amplifier Topologies, Analog Circuit Design, Machine Learning, Performance Parameters**

I. INTRODUCTION

Design automation for analog integrated circuits is still immature compared to the highly automated synthesis flow for digital circuits. We observe a severe bottleneck in analog VLSI design; the design of the analog parts of a mixed signal chip is usually more time-consuming than the design of the digital parts, even when the analog parts contain far less number of transistors and circuit blocks. This shows the need for new and better analog design methods. [1] Analog circuit design requires two steps: topology selection, which involves deciding the appropriate amplifier configuration for the given requirement and transistor sizing, which involves systematic configuration of the aspect ratio. Designing a CMOS analog amplifier is a process that requires a lot of knowledge and experience. The different characteristics of MOSFET contribute vividly to the performance of an amplifier. The designer must be mindful of various dependencies on parameters such as transconductance, power consumption, output current, voltage gain, unity gain frequency, phase margin, input noise, output noise and slew rate and carefully decide on transistor sizing. The numerous amplifier configurations (or amplifier topologies) also add to the dilemma of the designer. Conventionally, a designer is given a set of specifications and the designer is expected to design an amplifier that meets these requirements by choosing a suitable topology. This requires performing repeated tasks such as manual calculations and simulations. This process is tedious and sluggish. Hence, to overcome these issues, we have formulated a technique for selecting a suitable topology which meets the given specifications without performing multiple calculations and simulations, thereby, saving time of designers.

II. CMOS AMPLIFIER TOPOLOGIES AND PERFORMANCE PARAMETERS

A. Amplifier topologies

An amplifier is an electronic circuit that increases the power or voltage of a given input signal. There are many amplifier topologies built using CMOS technology such as common source with resistive load, common source with diode connected load, common drain, cascode amplifier, single stage OPAMP, two stage OPAMP, feedback amplifier, to name a few widely used topologies.

As common drain amplifier, cascode amplifier and 2 stage amplifiers form the basis of all complex amplifiers, these basic amplifier topologies have been chosen and demonstrated in this paper.

1) *Common drain:* The schematic is shown in Figure 1. Common drain amplifier, also known as source follower, is a single stage amplifier topology. This amplifier is typically used as a voltage buffer. In this topology (NMOS) the gate terminal of the transistor serves as the input, the source terminal is the output, and the drain terminal is common to both (input and output), hence its name. [2]
2) *Cascode amplifier:* The schematic is shown in Figure

2. The topology is designed by cascading the common source (CS) stage and common gate (CG) stage. The input is provided to the CG stage and hence generates a small signal drain current proportional to input. M1 is called input device and M2 as cascode device. [2]

3) *Two stage OPAMP:* The schematic is shown in Figure 3. Two stage OPAMP is an amplifier where 2 different amplifier topologies are cascaded. The pair can be high gain cascaded with high swing. Here the 1st stage consists of a single stage OPAMP whose output is cascaded to the input

of a simple common source amplifier to accommodate high swing. [2]

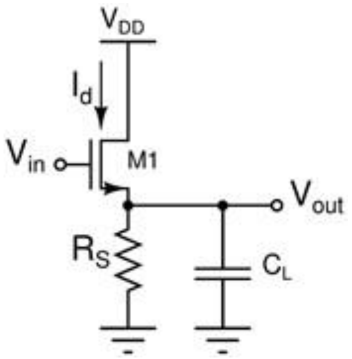

Fig. 1. Schematic of Common drain amplifier [3]

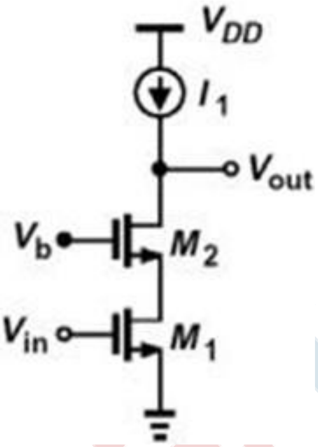

Fig. 2. Schematic of Cascode amplifier [2]

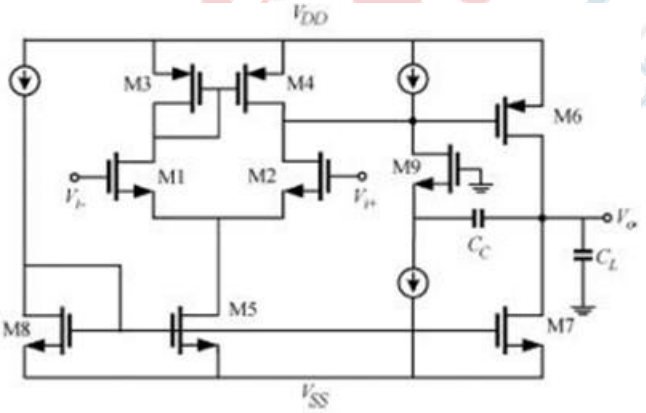

Fig. 3. Schematic of Two stage amplifier [4]

B. Performance parameters

The CMOS amplifiers are characterized by performance parameters, that can be extracted by performing various circuit analyses such as DC, AC, Transient and Noise Analysis.

1) *Transconductance:* Transconductance is the property of certain electronic components. Conductance is the reciprocal of resistance; transconductance is the ratio of the current change at the output port to the voltage change at the input port. It is written as ’gm’. [2]

2) *Maximum power consumption:* The static power consumption, or leakage power, is caused by leakage currents that are present in any active circuit, independently of clock rates and usage scenarios. This static power is mainly determined by the type of transistors and process technology. [2]

3) *Maximum output current:* Maximum output current of a CMOS amplifier is defined as the maximum current that can be supplied to the load. [2]

4) *Voltage gain:* Voltage gain of an amplifier can be interpreted as the ratio between the output voltage and the input voltage of the circuit. Using AC analysis, we can get gain in dB. [2]

5) *Unity gain frequency:* Gain and bandwidth in an amplifier are inversely proportional to each other and their relationship is summarized as the unity-gain bandwidth. Unity-gain bandwidth defines the frequency at which the gain of an amplifier is equal to 1 or 0 dB. The frequency corresponding to unity gain can be extracted from circuit simulations using frequency sweeps. [2]

6) *Phase Margin:* Phase margin is defined as the amount of change in open-loop phase needed to make a closed-loop system unstable. The greater the Phase Margin (PM), the greater will be the stability of the system. The phase margin refers to the amount of phase, which can be increased or decreased without making the system unstable. It is usually expressed as a phase in degrees. This is found by calculating the vertical distance between the phase curve (on the Bode phase plot) and the x-axis at the frequency where the Bode magnitude plot = 0 dB. This point is known as the gain crossover frequency. [2]

7) *Input noise:* Noise is an unwanted signal that creates disturbance to the desired signal content in the system. This can be an additional signal that is produced within the system or can be some disturbance accompanied by the desired information of the input signal. However, it is unwanted and has to be removed. The noise in the input signal is called input noise. [2]

8) *Output noise:* The noise in the output signal is called output noise. It is input noise multiplied by gain. [2]

9) *Slew rate:* Slew rate is defined as the maximum rate of change of an OPAMP’s output voltage and is given in units of volts per microsecond. Slew rate is measured by applying a large signal step, such as one volt, to the input of the OPAMP, and measuring the rate of change from 10% to 90% of the output signal’s amplitude. [2]

III. DESIGN FLOW

The design flow can be represented using a block diagram as shown in Figure 4.

A. Data Extraction

The Machine Learning techniques need a dataset for training the ML models. The data required to train the ML models for topology selection is obtained from the Cadence Virtuoso tool.

The data extraction procedure is as follows:

1) We use Cadence Virtuoso schematic editor to build the schematic of the amplifier. The supply voltage is provided or set as 1V and the length of all transistors is kept constant at 180nm to reduce the effects of channel length modulation. The process technology used is gpdk90. Resistance, bias voltages and widths of transistors are varied for performing a parametric sweep.

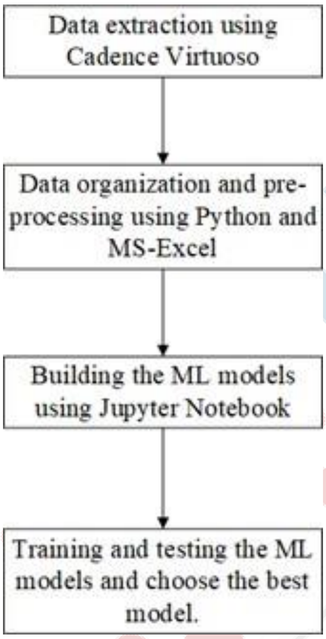
Data extraction using Cadence Virtuoso
Data organization and pre-processing using Python and MS-Excel
Building the ML models using Jupyter Notebook
Training and testing the ML models and choose the best model.

Fig. 4. Block diagram representation of design flow

2) Open ADE L and perform the following analyses:

-DC analysis to extract the values of transconductance, maximum output current and maximum power consumption.

-AC analysis to extract the values of voltage gain, unity gain frequency and phase margin.

-Transient analysis to extract slew rate.

-Noise analysis to extract input and output noise.

3) Perform a parametric sweep for the bias voltages, resistance and widths of transistors and extract the 9 performance parameters for different combinations of the sweep variables. Then the data is exported into a CSV file. Parametric Sweep is a procedure where one can vary different parameters w.r.t the topology in a given range to produce outputs for different combinations of input.

The above 3 steps are applied to all the 3 topologies (i.e., common drain, cascode amplifier and two stage OPAMP).

B. Data organization and pre-processing

After performing data extraction, we obtain 3 CSV files corresponding to the 3 topologies. The data of all the files are compiled into one CSV file that forms the dataset. We also need to annotate the dataset. Data annotation is the process of giving labels to the columns. Here we ensure we don't make any human errors. We also enter 'NaN' for missing values.

This completes the data organization and annotation process. The next step is data pre-processing. The following steps are performed for data pre-processing:

1) Loaded the dataset.
2) Removed the rows with duplicate entries.
3) For cascode amplifier and two-stage amplifier, the rows with negative values (in dB) for the voltage gain, because these circuits must have positive gain values (in dB) to operate as amplifiers. Retain the voltage gain values for common drain because this amplifier will only have negative voltage gain values (in dB). Perform log- transform on voltage gain values to convert them from dB to standard unit.
4) Checked 'NaN' values and substituting 0, because filling the 'NaN' values with 0 strongly affects the columns where 0 value is something impossible.
5) Checked if the dataset is balanced dataset or imbalanced dataset. Our dataset was found to be imbalanced, i.e., the number of data points for each class was different.
6) Before performing PCA, we drop the columns which are not required for the prediction of the topologies, i.e., we retain the columns that contain the 9 performance parameter values (called as feature columns) and the output label column that contains the topologies labeled as follows: 0 = Common drain, 1 = Cascode amplifier, 2 = Two stage OPAMP.
7) Then, we split the dataset into 2 parts: one consisting of feature columns and the other consisting of output label column, i.e., x-data and y-data respectively.
8) Found correlation between the feature columns and plotting it using heat map. But heat map is not a good criterion for finding out whether to drop a column or not. Hence, we use Principal Component Analysis for dimensionality reduction. After performing PCA, we found that all the feature columns should be retained.
9) Performed feature scaling. We performed min-max normalization for feature scaling.
10) Split x-data and y-data into x-train, x-test, y-train and y-test, i.e., splitting into training and testing sets for feature columns and output label column. The splitting ratio is 70% training and 30% testing.

C. Building, training and testing the ML models

Since our method deals with topology selection or prediction, we need to build classification algorithms. The potential classification algorithms must fulfill the following requirements:

1) They should work well for small/medium-sized datasets.
2) They should work well for multi-classification problems.
3) They should work well for numeric data.
4) Building them should be least expensive and should take less time.
5) Will potentially give good accuracy/precision/F1 score. Keeping the above considerations in mind, we decided to implement the following 5 ML models:
   1) *Logistic Regression:* Logistic Regression is a supervised learning algorithm which is used for binary classification. Generally, logistic regression is well suited for describing and testing hypotheses about relationships between a categorical outcome variable and one or more categorical or continuous predictor variables. [5]
   2) *K-Nearest Neighbors:* K-Nearest Neighbor (KNN) algorithm is an effortless but productive machine learning algorithm. It is effective for classification as well as regression. However, it is more widely used for classification prediction. KNN groups the data into coherent clusters or subsets and classifies the newly inputted data based on its similarity with previously trained data. The input is assigned to the class with which it shares the most nearest neighbors. [6]
   3) *Support Vector Classifier:* The support vector machine/classifier is a new type of machine learning method based on statistical learning theory. Because of good promotion and a higher accuracy, support vector machine has become the research focus of the machine learning community. [7] SVM map input vector to a higher dimensional space where a maximal separating hyperplane is constructed. Two parallel hyperplanes are constructed on each side of the hyperplane that separate the data. The separating hyperplane is the hyperplane that maximizes the distance between the two parallel hyperplanes. An assumption is made that the larger the margin or distance between these parallel hyperplanes, the better the generalization error of the classifier will be. [8]
   4) *Decision Tree Classifier:* A normal tree includes root, branches and leaves. The same structure is followed in Decision Tree. It contains root node, branches, and leaf nodes. Testing an attribute is on every internal node, the outcome of the test is on branch and class label as a result is on leaf node. [9] [10] A root node is parent of all nodes and as the name suggests it is the topmost node in tree. A decision tree is a tree where each node shows a feature (attribute), each link (branch) shows a decision (rule), and each leaf shows an outcome (categorical or continues value). [10]
   5) *Random Forest Classifier:* Random forests are a com- bination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. [11] RF follows specific rules for tree growing, tree combination, self-testing and post-processing, it is robust to overfitting, and it is considered more stable in the presence of outliers and in very high dimensional parameter spaces than other machine learning algorithms. [12]
   6) The model building process is common for all the 5 algorithms. The process is shown in Figure 5.

IV. RESULTS

The performance metrics which we get after training and testing the ML models are shown in Table I. Since our dataset is imbalanced, we find precision, recall and F1 score for test data. The heat maps of the results for the 5 ML algorithms are shown in Figures 6, 7, 8, 9 and 10.

V. DISCUSSION

The results of the study indicate that K-Nearest Neighbors algorithm performs the best overall. It performs well in terms of training which indicates that it has not underfit the data and performs the best in terms of testing, which indicates it has not overfit the data.

The purpose of the study was to formulate a new method of topology selection which reduces the stress on a designer who would have to brainstorm about the specifications given by the client to decide which topology is suitable, but now only has to provide these parameters to the ML model and get the suitable topology. The results of the study confirm that ML based topology selection is efficient, fast and accurate. However, there is a drawback to this method. ML algorithms require a dataset which might not be readily available. The extraction of data is time-consuming and cumbersome and hence it is a challenge while using ML based methods. But this can be easily overcome if we have readily available data or if we have computers with increased computational power to speed up the extraction process. Therefore, ML based algorithms are a good choice for topology selection as compared to age-old methods such as pool selection or genetic algorithm, which have little acceptance by industrial designers.

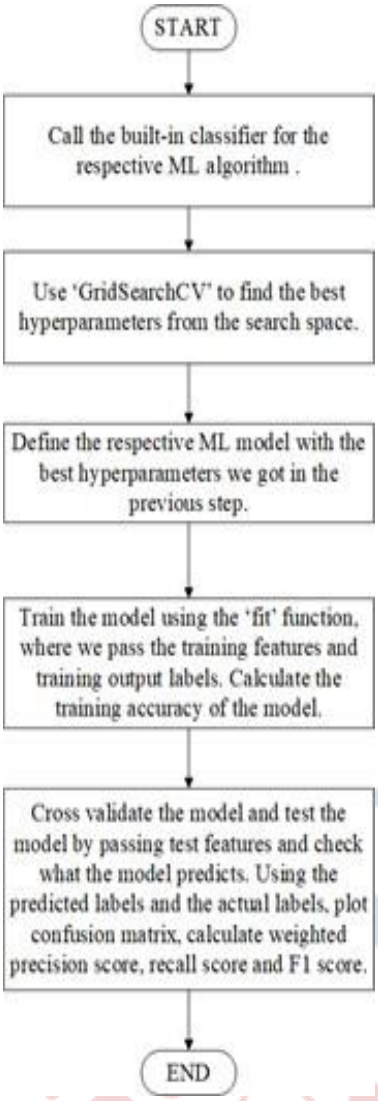
START
Call the built-in classifier for the respective ML algorithm .
Use 'GridSearchCV' to find the best hyperparameters from the search space.
Define the respective ML model with the best hyperparameters we got in the previous step.
Train the model using the 'fit' function, where we pass the training features and training output labels. Calculate the training accuracy of the model.
Cross validate the model and test the model by passing test features and check what the model predicts. Using the predicted labels and the actual labels, plot confusion matrix, calculate weighted precision score, recall score and F1 score.
END

Fig. 5. ML model building algorithm

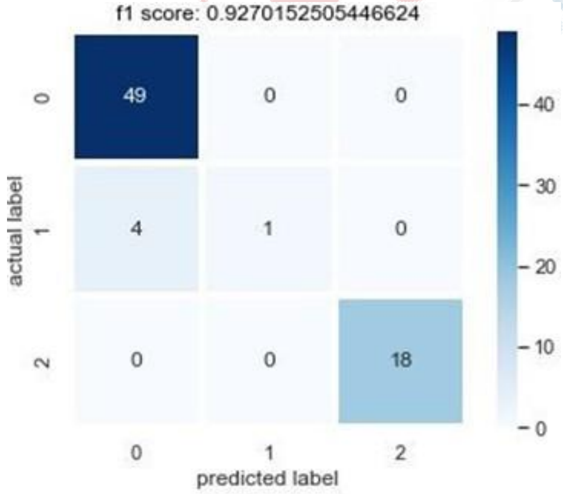

Fig. 6. Confusion matrix heat map of Logistic Regression model on test data

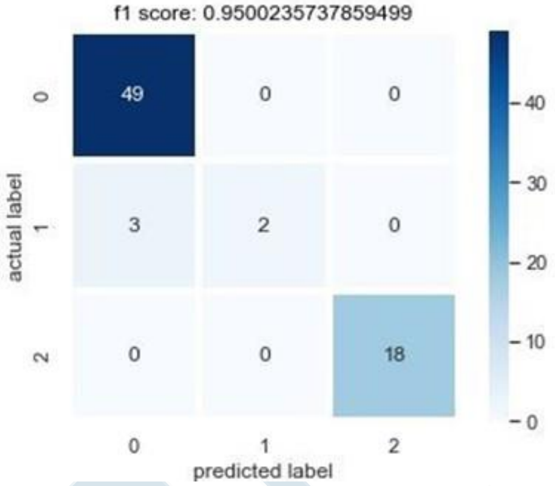

Fig. 7. Confusion matrix heat map of K-Nearest Neighbors model on test data

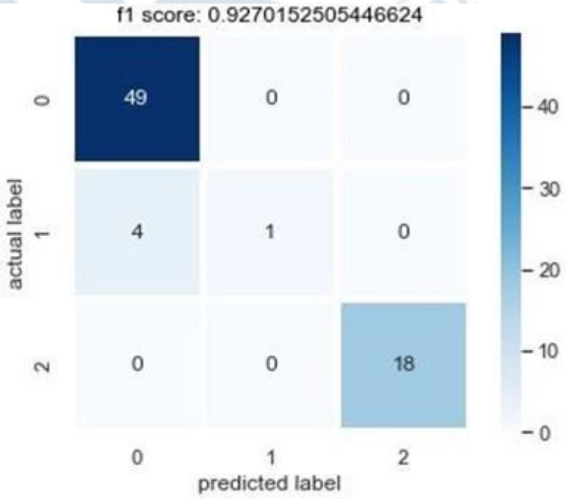

Fig. 8. Confusion matrix heat map of Support Vector Classifier model on test data

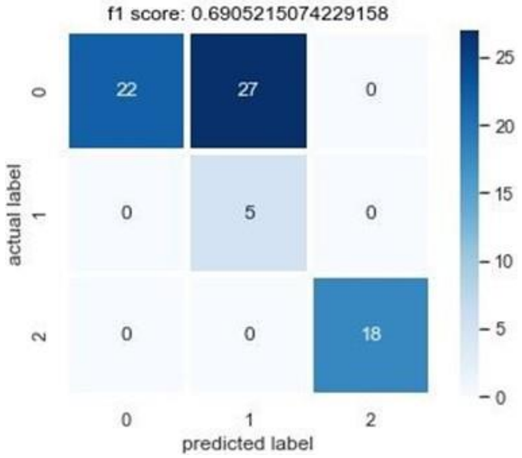

Fig. 9. Confusion matrix heat map of Decision Tree Classifier model on test data

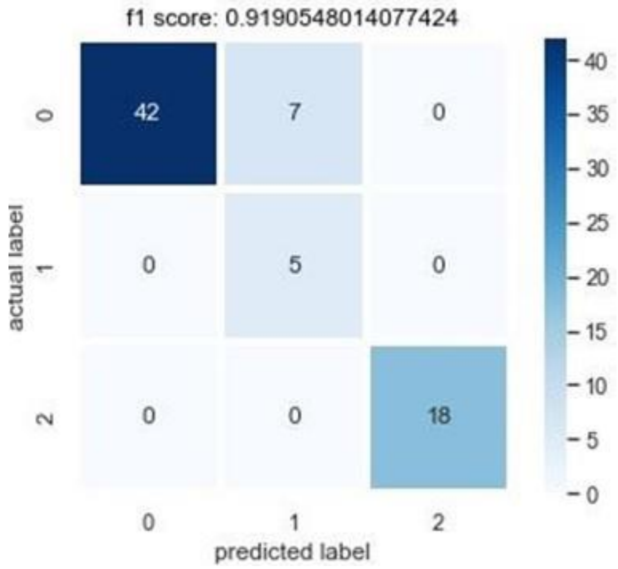

Fig. 10. Confusion matrix heat map of Random Forest Classifier model on test data

VI. CONCLUSION

Topology selection is an integral part of circuit design and hence choosing the right topology is important. The design engineers take a lot of time to comprehend and choose the right topology which could potentially meet all the specifications given by the client. This is a difficult and a repetitive process. In order to overcome the above issues, we have proposed a topology selection method which employs an ML algorithm which is faster, more reliable and efficient. We performed a study on topology selection using 5 different ML algorithms and compared the performances of each and deduced that K- Nearest Neighbors algorithm was the best model.

TABLE I TRAINING AND TESTING PERFORMANCE METRICS

ML Model	Training accuracy	Test Precision	Test Recall	Test F1 Score
Logistic Regression	99%	94.86%	94.44%	92.70%
K-Nearest Neighbors	99%	96.07%	95.83%	95.00%
Support Vector Classifier	99%	94.86%	94.44%	92.70%
Decision Tree Classifier	99%	94.14%	62.50%	69.05%
Random Forest Classifier	100%	95.95%	90.27%	91.90%

REFERENCES

[1] A. Gerlach, T. Rosahl, F. T. Eitrich, and J. Scheible, “A generic topology selection method for analog circuits with embedded circuit sizing demonstrated on the OTA example,” in Conf. for Design and Test in Europe, 2017, pp. 898–901.

[2] B. Razavi, “Design of Analog CMOS Intergrated Circuits.”

[3] P. Bhattacharjee, A. J. Mondal, and A. Majumder, “A constraint driven technique for MOS amplifier design,” 2016.

[4] A. Sadeqi, J. Rahmani, S. Habibifar, M. A. Khan, and H. M. Munir, “Design method for two-Stage CMOS operational amplifier applying load/miller capacitor compensation,” Computational Research Progress in Applied Science & Engineering, vol. 06, no. 03, pp. 153–162, 2020.

[5] J. Peng, K. L. Lee, and G. M. Ingersoll, “An Introduction to Logistic Regression Analysis and Reporting,” Journal of Educational Research- J EDUC RES, vol. 96, 2002.

[6] K. Taunk, S. De, S. Verma, and A. Swetapadma, “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification,” 2019 International Conference on Intelligent Computing and Control Systems (ICCS), pp. 1255–1260, 2019.

[7] Y. Zhang, L. C., W. L., and Y. A., “Support Vector Machine Classification Algorithm and Its Application,” in Information Computing and Applications. ICICA 2012, vol. 308. Springer, 2012.

[8] V. N. Vapnik, “The Nature of Statistical Learning Theory,” 2000.

[9] A. Gershman, A. Meisels, K. Lüke, L. Rokach, A. Schclar, and A. Sturm, “A Decision Tree Based Recommender System,” 2010.

[10] S. D. Jadhav and H. Channe, “Efficient Recommendation System Using Decision Tree Classifier and Collaborative Filtering,” 2016.

[11] L. Breiman, “Random Forests,” Machine Learning, vol. 45, 2001.

[12] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in 23rd International Conference on Machine Learning. ACM Press, 2006.